

Bachelor-Thesis

in

Medieninformatik

Konzeption, Implementation und Evaluation von Netzwerkkomponenten für eine auf die Lehre spezialisierte Game-Engine

Referent: Prof. Jirka Dell‘Oro-Friedl

Korreferent: Prof. Dr. Ruxandra Lasowski

Vorgelegt am: 31.8. 2100

Vorgelegt von: Falco Böhnke

250100

Im Großacker, 28

79252, Stegen

falco.boehnke@hs-furtwangen.de

Abstract 100-120wort

Im Rahmen dieser Bachelorarbeit wurden Komponenten für die Netzwerkkommunikation innerhalb des auf Electron basierenden Editors „Fudge“ konzipiert und entwickelt. Dafür werden zuerst die Anforderungen, vorgegeben durch Electron und Fudge selbst, festgelegt.

Im Anschluss daran wird erläutert, welche Varianten von Netzwerkkommunikation in heutigen Videospielen üblich sind, welche Protokolle für die Datenübertragung verfügbar sind und welche existierenden Webtechnologien verfügbar sind um mithilfe der jeweiligen Protokolle zu kommunizieren.

Nach Auswahl der zu verwendenden Technologien wird die Konzeption erläutert und die Komponenten umgesetzt und ihre Funktionsweise mithilfe von UML-Diagrammen und Aktivitäts-Diagrammen dargelegt.

Zum Abschluss wird auf die Zukunftsaussichten und etwaige Möglichkeiten der Weiterentwicklung eingegangen und eine Schlussfolgerung über die Entwicklung von Netzwerkkomponenten für einen didaktischen Spiele gezogen.

Inhaltsverzeichnis

[Abstract 3](#_Toc17290126)

[Inhaltsverzeichnis 5](#_Toc17290127)

[Abbildungsverzeichnis 7](#_Toc17290128)

[Abkürzungsverzeichnis 9](#_Toc17290129)

[Einleitung 1](#_Toc17290130)

[Fudge – Didaktischer Spieleeditor 3](#_Toc17290131)

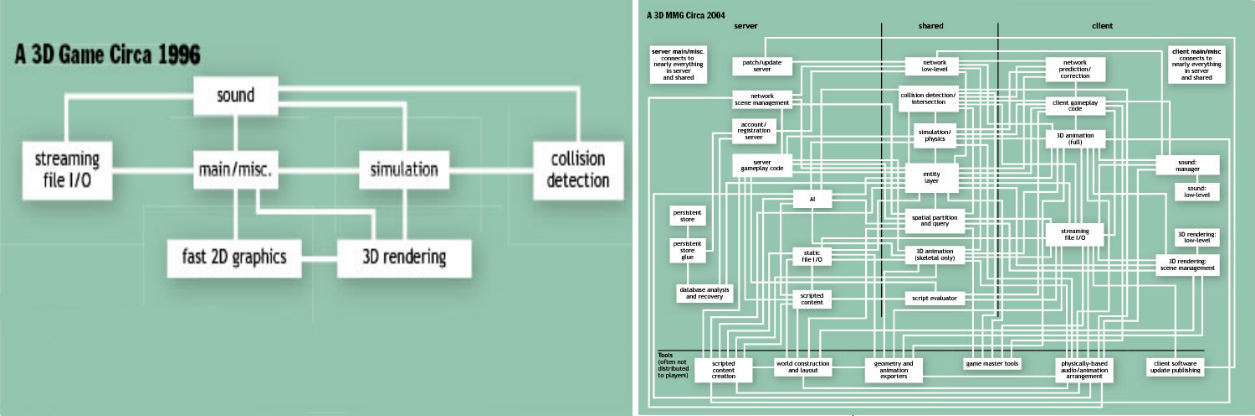
Abbildungsverzeichnis

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Abkürzungsverzeichnis

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

1 Einleitung

Die Videospielindustrie ist in den letzten Jahren massiv gewachsen. Allein der Markt für Videospiele in den Niederlanden wuchs von 518 Millionen Euro im Jahr 2012 auf 1.65 Milliarden Euro im Jahr 2019 an, mit vergleichbaren Tendenzen, mit Wachstumszahlen von 11% über die letzten 10 Jahre, weltweit (Wijman 2018) (Kazimier 2017). Der starke Anstieg der Umsatzzahlen hat zu einem gesteigerten Konkurrenzkampf innerhalb des Videospielmarktes geführt. Innovationen sind unerlässlich geworden und so sehen sich Entwickler mit immer komplexeren Projektanforderungen konfrontiert. Insbesondere im Bereich der Netzwerkkommunikation sind die Anforderung an hoch performante Lösungen gestiegen.

**Abb. 1:** Vergleich der möglichen Komponenten für Videospiele aus den Jahren 1996 und 2004 in Anlehnung an Blow (2004)

Gameengines, Entwicklungsumgebungen, die für die Produktion von Videospielen optimiert sind, finden daher verstärkt Einsatz um Entwickler die Arbeit zu erleichtern und komplexe Komponenten verwendungsfertig bereit zu stellen. Diese fertigen Komponenten sollen durch Automation und Abstraktion Entwicklern fundamentale Arbeiten abnehmen und so Kapazitäten für Optimierung befreien. Doch die Undurchsichtigkeit der abstrahierten Systeme und die oftmals komplizierte Struktur der Game-Engines macht es Entwicklern in Ausbildung schwer die grundlegende Funktionsweise der einzelnen Komponenten eines Videospiels zu erfassen und zu erlernen. So sind Entwickler gezwungen statt grundlegender Konzepte die Funktionsweise ihrer ausgewählten Entwicklungsumgebung zu lernen. Ist dann ein Wechsel auf eine andere Entwicklungsumgebung notwendig, müssen Entwickler sich erneut an die Umgebungsentwicklung anpassen, wobei dort ähnliche Komponenten mitunter andere Funktionsweisen haben. Besonders der Mangel einer von Grund auf für die Lehre konzipierten Game-Engine verschärft diese Problematik weiter.

Um diese Lücke zu füllen wurde das Projekt ‚Fudge‘ von Prof. Jirka Dell‘Oro-Friedl ins Leben gerufen. Fudge ist eine Game-Engine und ein Editor, der die Strukturen und Prozesse eines Videospiels offen legt und Entwickler dennoch mit grundlegenden Funktionalitäten versorgt.

In dieser Arbeit wurden Netzwerkkomponenten entwickelt, die Entwicklern in Fudge die Möglichkeit geben vernetzte Spiele zu entwickeln, ohne die notwendigen Komponenten von Grund auf selbst zu schreiben. Dazu wird zuerst der Entwicklungsrahmen mit Fudge und Electron dargelegt. Anschließend wurden die in modernen Spielen üblichsten Methoden zur Kommunikation zwischen Spielinstanzen ermittelt und kurz erläutert. Dazu gehört ebenfalls ein kurzer Exkurs in modernen Netzwerkprotokollen die in der Netzwerkkommunikation üblich sind und daher relevant für die Umsetzung der Netzwerkkommunikation in Fudge.

Aus diesen Informationen erschließen sich die Anforderungen an die Netzwerkkomponenten und die Aufgaben die sie erfüllen können müssen, um für die Entwicklung von Spielen in Fudge sinnvoll verwendet werden zu können.

Unter Berücksichtigung der Rahmenbedingungen und Anforderungen werden dann die möglichen Technologien ermittelt, die bei der Entwicklung verwendet werden können. Diese werden evaluiert und die passenden werden im weiteren für die Entwicklung verwendet

Darauf folgend wird die Umsetzung der einzelnen notwendigen Netzwerkkomponenten dargelegt und ihr Aufbaue mithilfe von Unified Modeling Language Diagrammen dargestellt, sowie ihre Funktionsweise und Ablauf beispielhaft mithilfe von Aktivitätsdiagrammen dargestellt.

Abschließend werden die Ergebnisse der Evaluation und Entwicklung zusammengefasst und ein Fingerzeig für eine mögliche Weiterentwicklung der Komponenten gegeben.

2 Rahmenbedingungen – Theoretische Grundlagen

Fudge

Typescript/Javascript

Electron

Nodejs

Bla

Bla

Netzwerkprotokolle

Udp

Tcp

ftp

Heutige Spiele

Clientserver

Peertopeer

3 Methodik

Im letzten kapitel wurden die Voraussetzungen dargelegt die Technologien erfüllen müssen um für die Entwicklung von Netzwerkomponenten in Frage zu kommen. Im folgenden werden die möglichen Kandidaten vorgestellt und auf ihre Eignung geprüft.

Auswahl der kompatiblen Technologien

Darstellung als grafik zeitstrahl wo inkompatible sachen dann runterfallen sozusagen oder eher plus minus rot grün tabelle

Tech 1

Tech 2

Tech 3

Entwicklung

Komponenten

Komp 1

Komp 2

Komp 3

Komp 4

4. Praktische Anwendung

Neue Nachrichten abfangen

Clientenlogik einbauen

Authoritative Commands strukturieren

Ids für netzwerkobjete vergeben

5. Diskussion und Ausblick

5.1 Diskussion

5.2 Weiterentwicklung der Kernkomponenten

Wie können die komponenten selbst weiterentwickelt werden z.b, durch

**server prediction,**

**lazy loading,**

**nachrichten validierung,**

**werte validierung**